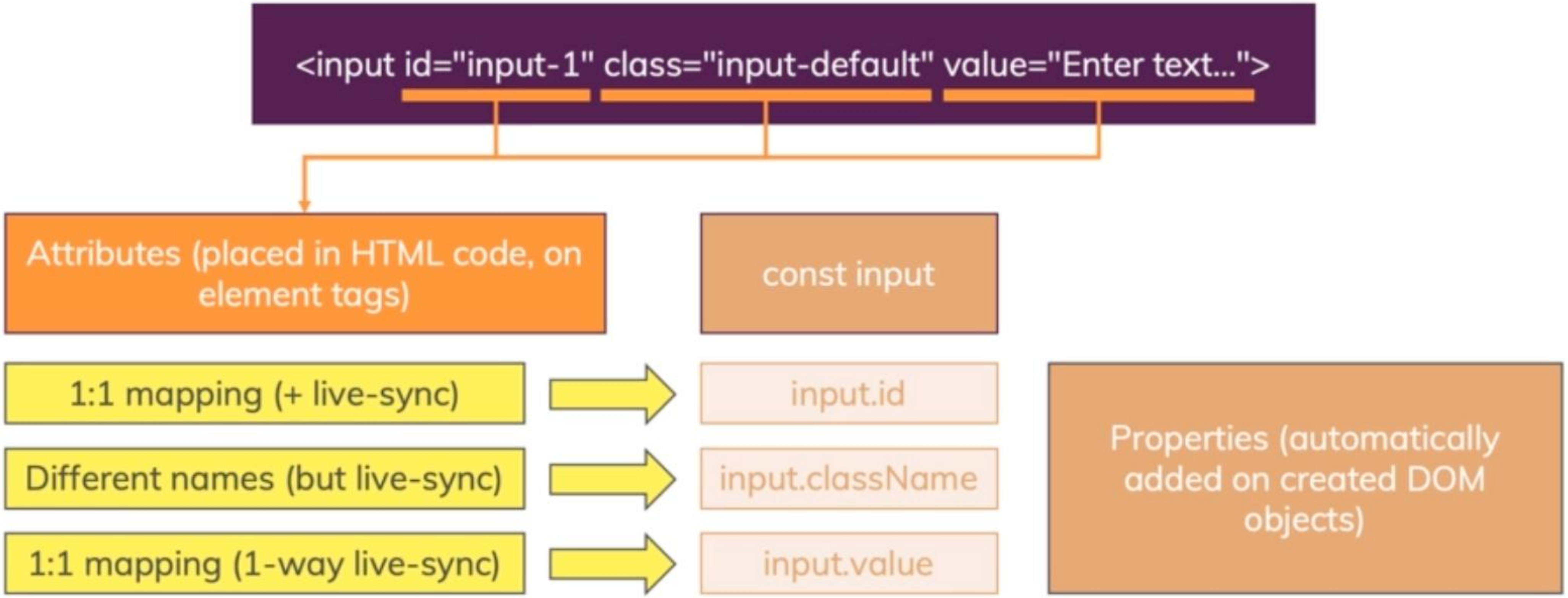
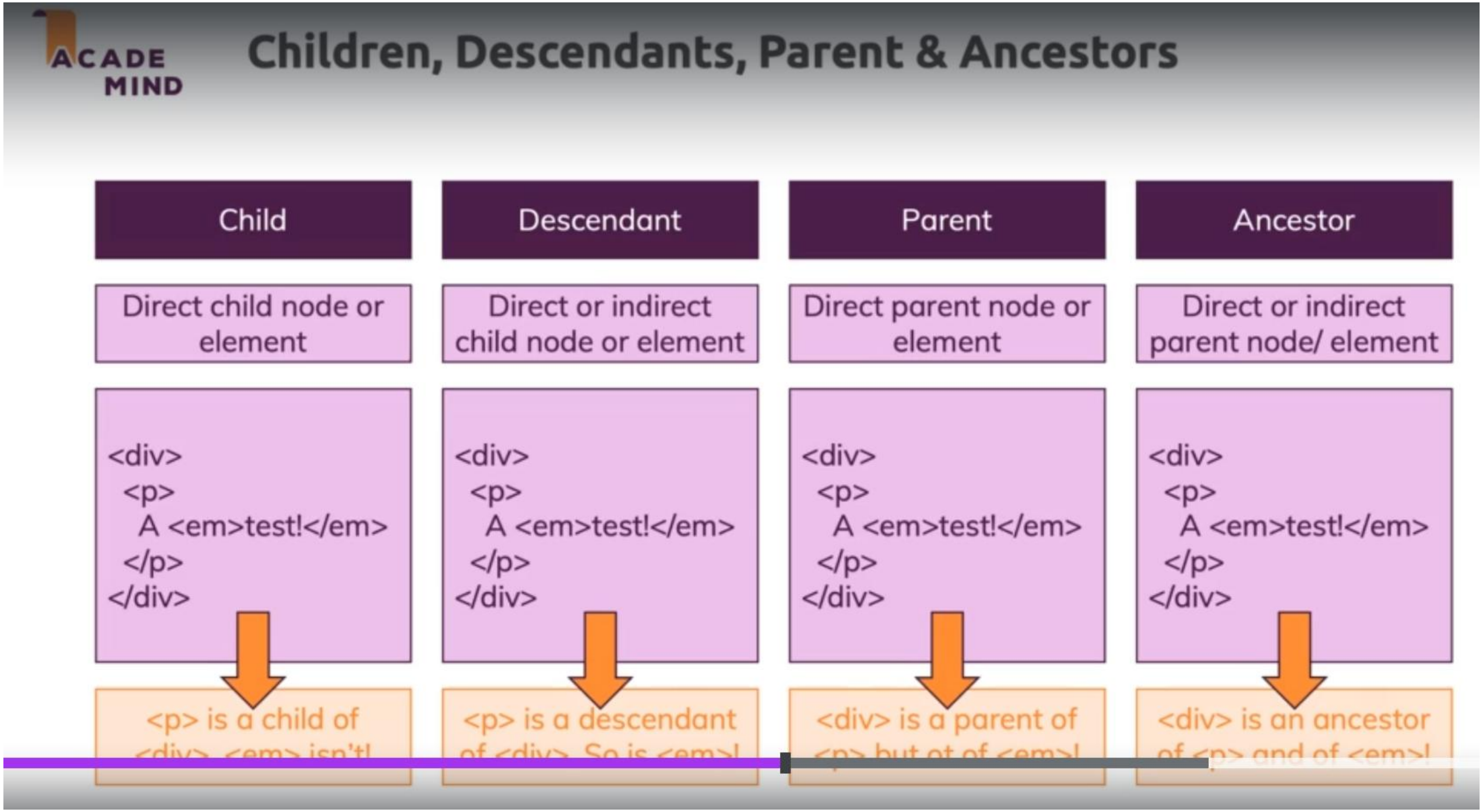
Attributes vs Properties

Often (but not always!), attributes are mapped to properties and “live synchronization“ is set up.



```
<input id="input-1" class="input-default" value="Enter text...">
```

Attributes (placed in HTML code, on element tags)
const input
1:1 mapping (+ live-sync)
input.id
Different names (but live-sync)
input.className
1:1 mapping (1-way live-sync)
input.value
Properties (automatically added on created DOM objects)



Children, Descendants, Parent & Ancestors

Child

Direct child node or element

```
<div>
  <p>
    A <em>test!</em>
  </p>
</div>
```

<p> is a child of <div>, <em> isn't!

Descendant

Direct or indirect child node or element

```
<div>
  <p>
    A <em>test!</em>
  </p>
</div>
```

<p> is a descendant of <div>. So is <em>!

Parent

Direct parent node or element

```
<div>
  <p>
    A <em>test!</em>
  </p>
</div>
```

<div> is a parent of <p> but ot of <em>!

Ancestor

Direct or indirect parent node/ element

```
<div>
  <p>
    A <em>test!</em>
  </p>
</div>
```

<div> is an ancestor of <p> and of <em>!

ACADE MIND

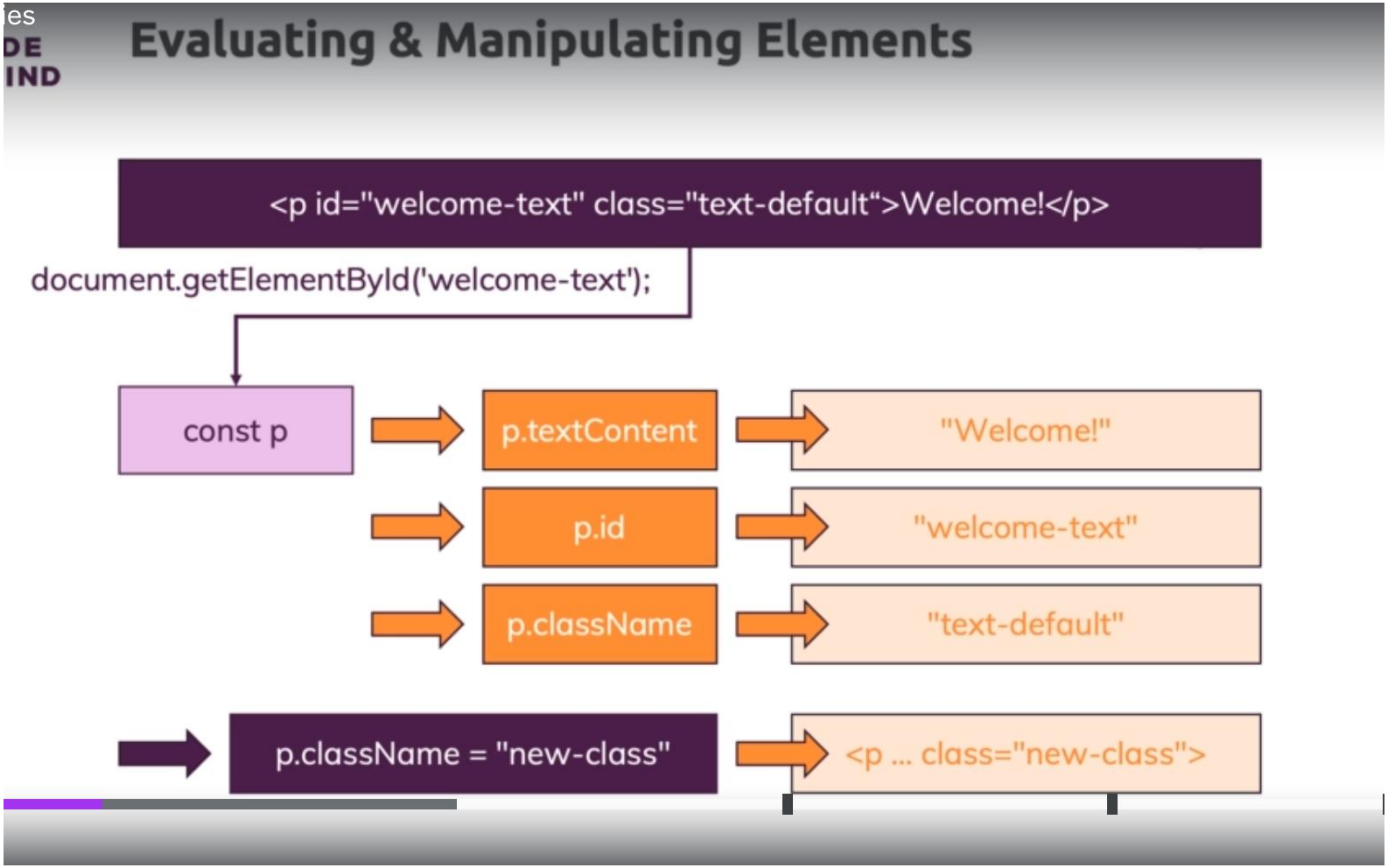
Creating & Inserting Elements

HTML string	innerHTML	Add (render) HTML string
	insertAdjacentHTML()	Add (render) HTML string in specific position
createElement()	appendChild() / append()	Append new DOM element/ node
	prepend(), before(), after(), insertBefore()	Insert new DOM element/ node in specific position
	replaceChild(), replaceWith()	Replace existing DOM element/ node with new one

document & window

document	window
Root DOM Node	The active Browser Window / Tab
Provides access to element querying, DOM content etc	Acts as global storage for script, also provides access to window-specific properties and methods

Evaluating & Manipulating Elements



<p id="welcome-text" class="text-default“>Welcome!</p>
document.getElementById('welcome-text');
const p
p.textContent
"Welcome!"
p.id
"welcome-text"
p.className
"text-default"
p.className = "new-class"
<p ... class="new-class">

Insertion & Removal Methods

- appendChild()
- insertAdjacentElement()
- replaceChild()
- removeChild()

Broad browser support

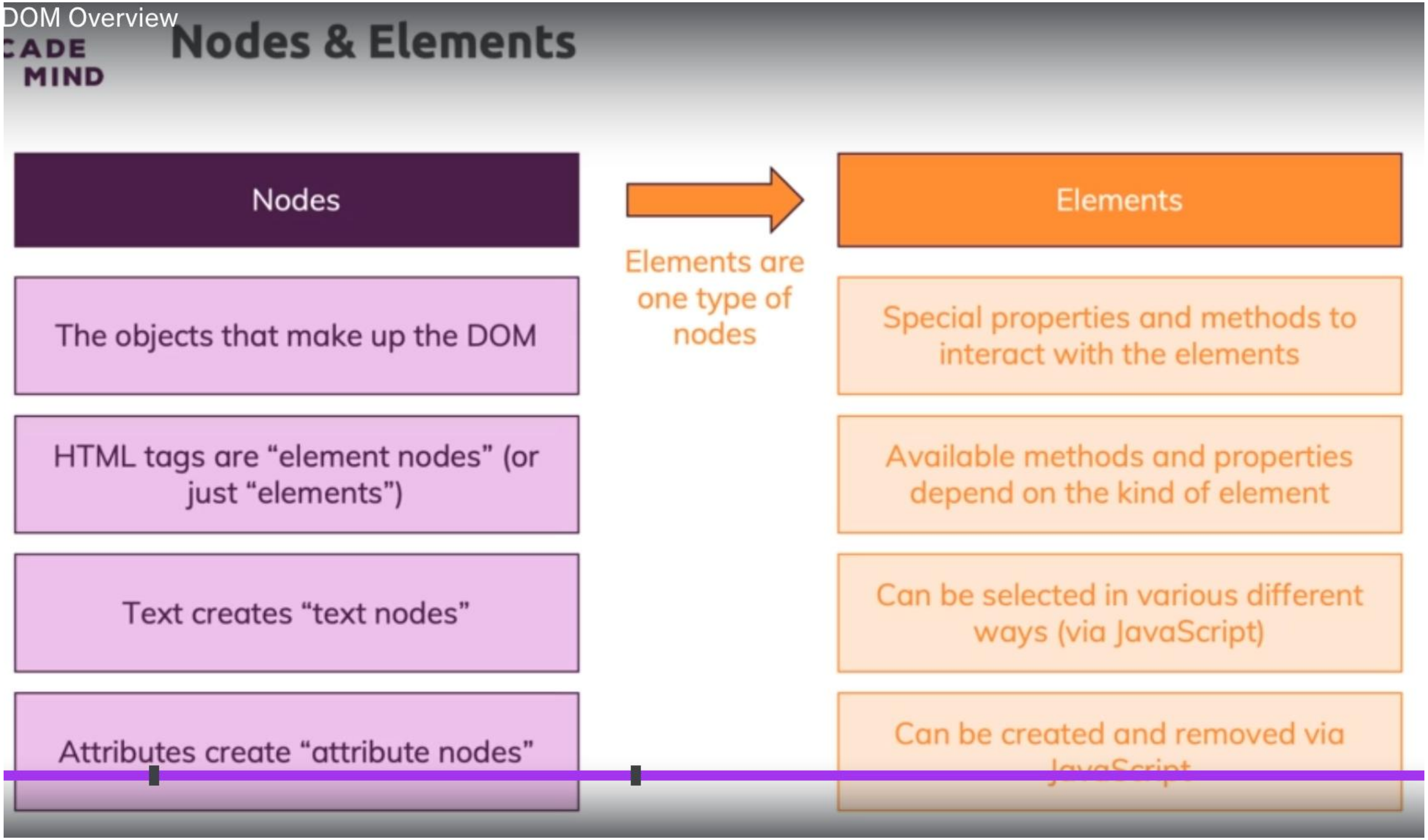
Plenty of resources on the internet

NOT supported in all browsers

Seen in some tutorials but not most common option

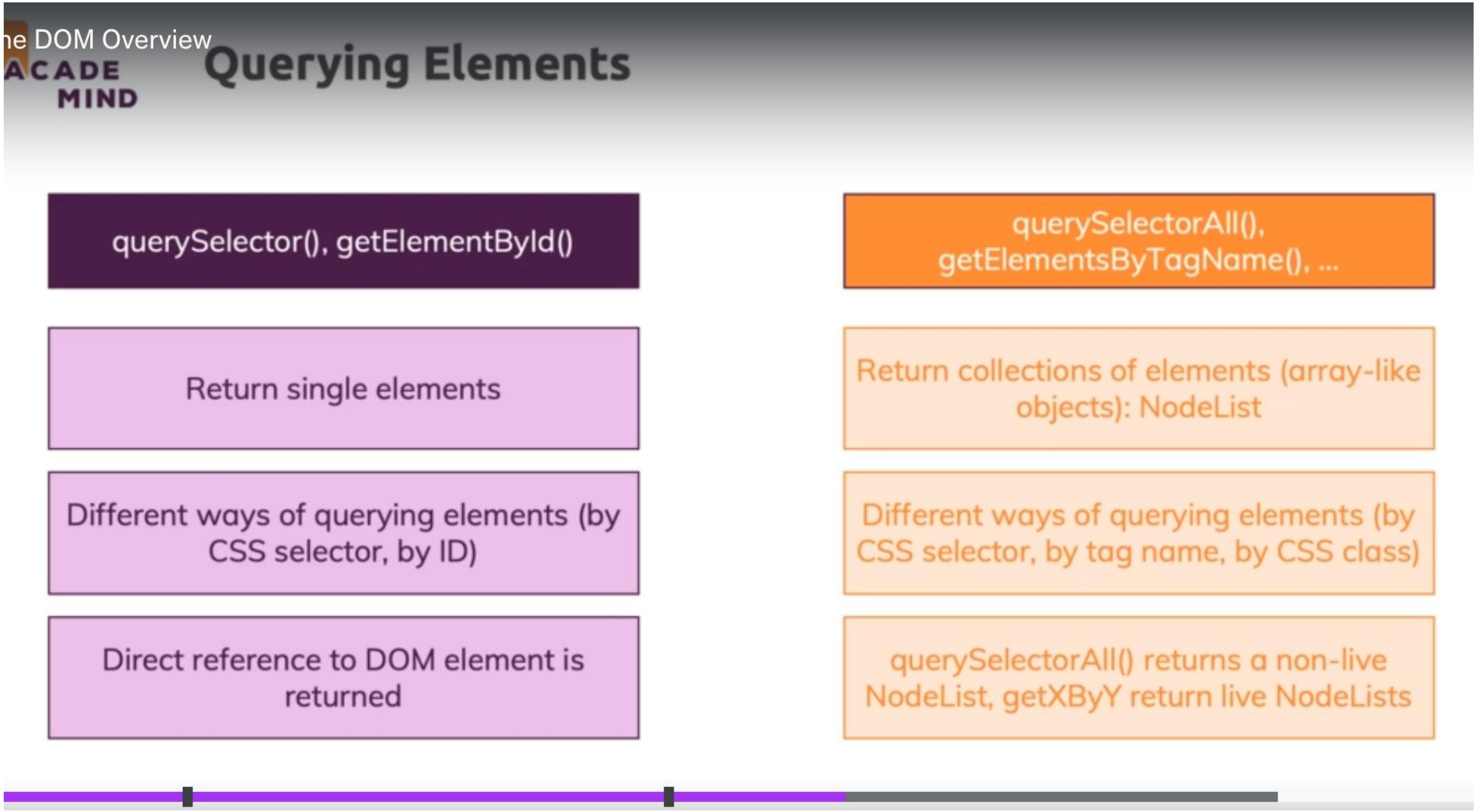
- append()
- prepend()
- before()
- after()
- replaceWith()
- remove()

DOM Overview



Nodes & Elements

Nodes		Elements
The objects that make up the DOM	Elements are one type of nodes	Special properties and methods to interact with the elements
HTML tags are "element nodes" (or just "elements")		Available methods and properties depend on the kind of element
Text creates "text nodes"		Can be selected in various different ways (via JavaScript)
Attributes create "attribute nodes"		Can be created and removed via JavaScript



he DOM Overview

Querying Elements

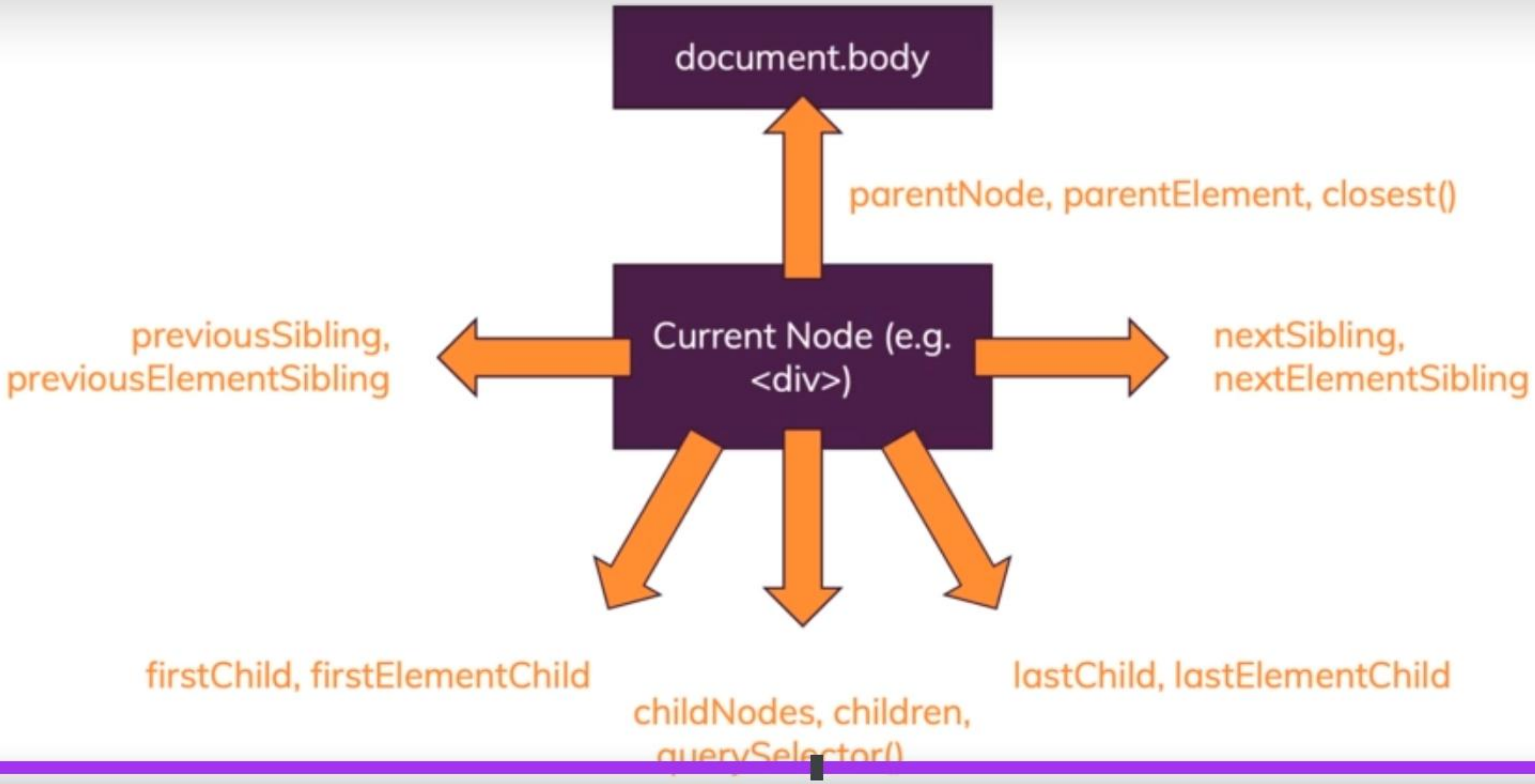
querySelector(), getElementById()	querySelectorAll(), getElementsByTagName(), ...
Return single elements	Return collections of elements (array-like objects): NodeList
Different ways of querying elements (by CSS selector, by ID)	Different ways of querying elements (by CSS selector, by tag name, by CSS class)
Direct reference to DOM element is returned	querySelectorAll() returns a non-live NodeList, getXByY return live NodeLists

ACADE MIND

Styling DOM Elements

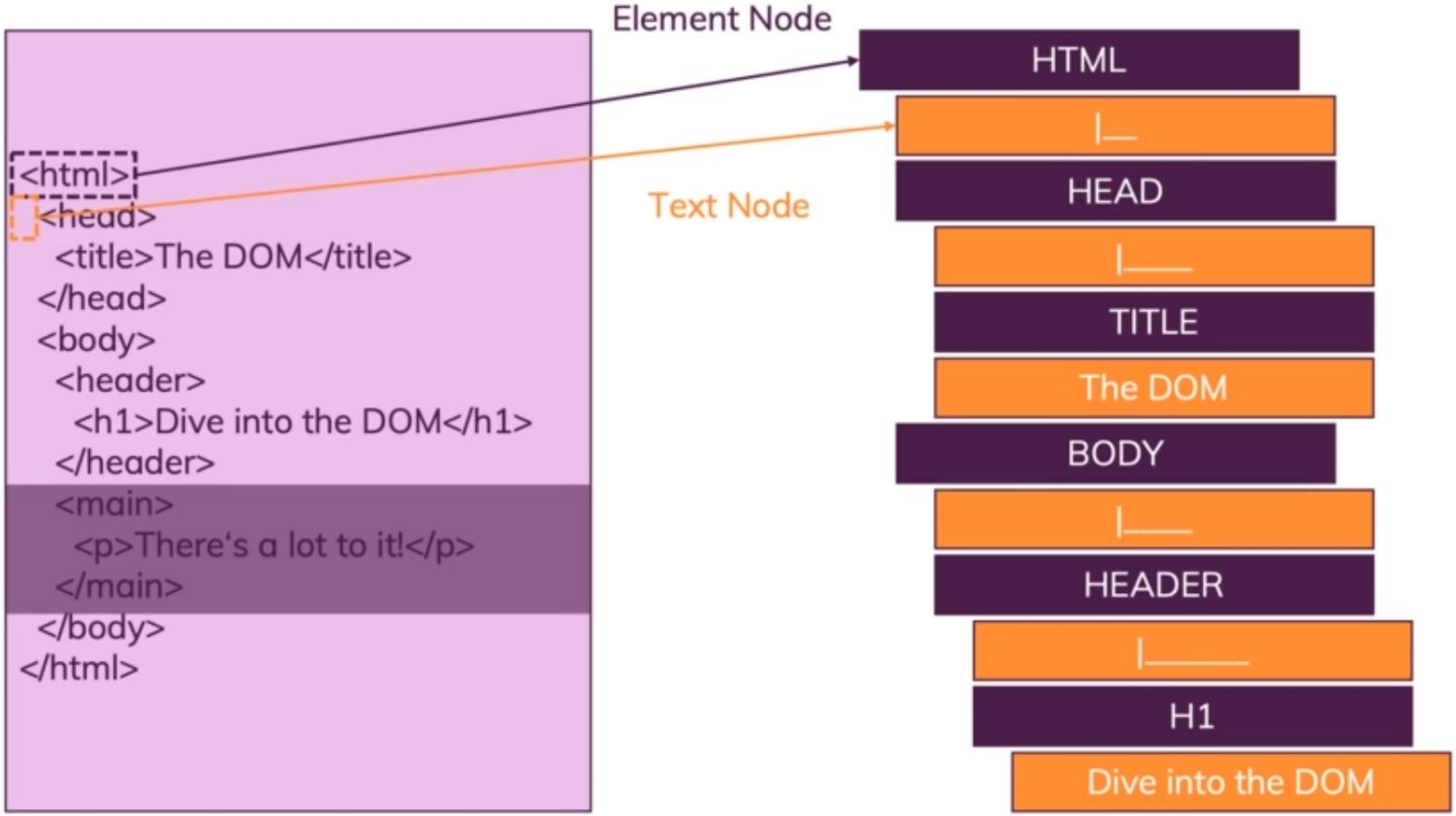
Via style Property	Via className	Via classList
Directly target individual CSS styles (on the element)	Directly set the CSS classes assigned to the element	Conveniently add, remove or toggle CSS classes
Controls styles as inline styles on the element	Set/ Control all classes at once	Fine-grained control over classes that are added
Style property names are based on CSS properties but have adjusted names (e.g. backgroundColor)	You can also control the id or other properties	Can be used with className (with care)

Traversing the DOM



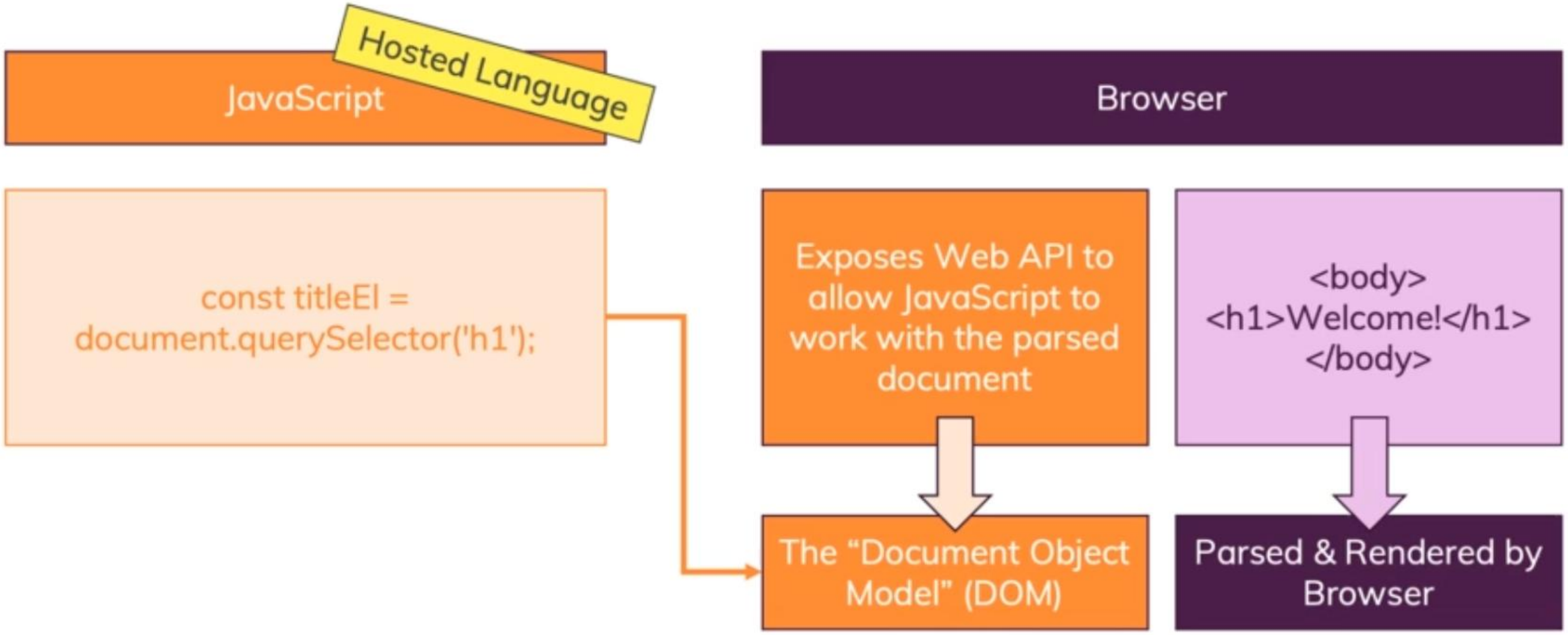
document.body
parentNode, parentElement, closest()
Current Node (e.g. <div>)
previousSibling, previousElementSibling
nextSibling, nextElementSibling
firstChild, firstElementChild
childNodes, children, querySelector()
lastChild, lastElementChild

The Document Object Model (DOM)



```
<html>
 <head>
  <title>The DOM</title>
 </head>
 <body>
  <header>
   <h1>Dive into the DOM</h1>
  </header>
  <main>
   <p>There's a lot to it!</p>
  </main>
 </body>
</html>
```

The Document Object Model (DOM)


JavaScript
Hosted Language
Browser
const titleEl =
document.querySelector('h1');
Exposes Web API to allow JavaScript to work with the parsed document
<body>
<h1>Welcome!</h1>
</body>
The "Document Object Model" (DOM)
Parsed & Rendered by Browser

DOM is not strictly tied to Browsers! There are other tools that can parse HTML!